Stand-alone PGAP: an NCBI open-source pipeline to annotate prokaryotic genomes

F. Thibaud-Nissen*, D. Slotta, A. Badretdin, B. Busby, R. Cohen, W. Li, W. Hlavina
National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD, 20892
*thibaudf@nih.gov

Download PGAP: https://github.com/ncbi/pgap/

Come by the **NCBI booth, #433**

We will talk about PGAP
Sat June 22 12:30-12:45 PM

Abstract

The NCBI Prokaryotic Genome Annotation Pipeline (PGAP), which has been used to annotate RefSeq prokaryotic genomes since the early 2000s, has increased in quality and consistency over the years. PGAP annotation, also offered as a service to researchers submitting genome assemblies to GenBank, has become a reliable resource for the prokaryotic community.

We have re-factored PGAP into a stand-alone pipeline that can be executed outside of NCBI on individual computers or in a cloud environment, with the goal of producing annotation that is in line with internal NCBI PGAP and that conforms to GenBank's standards of quality and format.

The pipeline is written in the Common Workflow Language (CWL) and is packaged with the necessary binaries and cwltool, the CWL reference implementation. All necessary reference data, including a variety of manually curated evidence and other datasets, are bundled and distributed with the pipeline. As a result, the annotations produced by stand-alone PGAP conform with annotations generated at NCBI.

We have recently simplified the user-provided inputs to a minimal set of files (FASTA for the genomic sequence and YAML for the metadata) and have added to the pipeline quality and consistency checks of the input data. In addition, the output is formatted for submission to GenBank, and, given sufficient metadata on input, the stand-alone PGAP annotation results can be submitted to GenBank with the assembly.

We expect that making PGAP portable will accelerate research by providing scientists quality annotations of the genomes they assemble prior to submission. It will also allow users to iterate over the assembly process until the annotation results demonstrate the quality of the assembly.

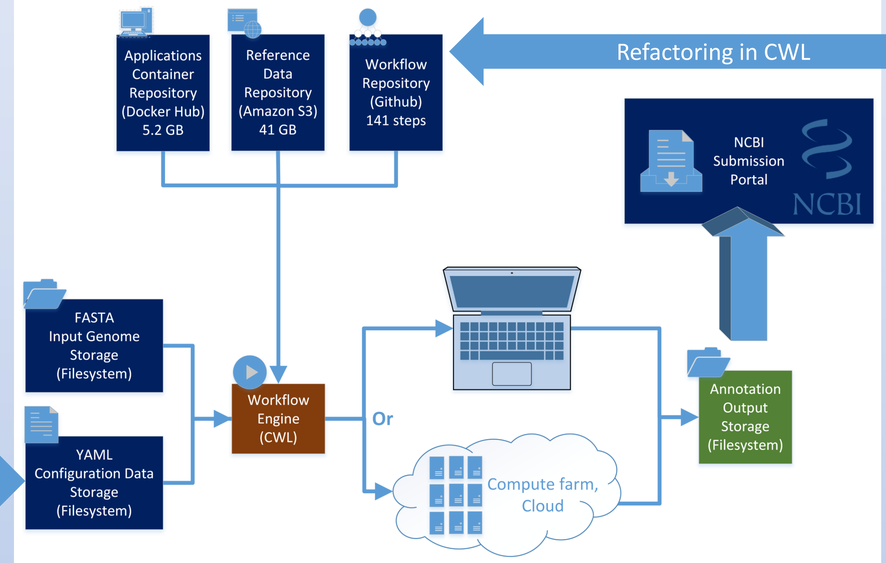
Example input file

The metadata for the genome to annotate is provided by the user in a YAML file, and is included in the output so results can easily be submitted to GenBank.

```
topology: 'circular'
organism:
    genus_species: 'Escherichia coli'
    strain: 'my_strain'
contact_info:
    last_name: 'Doe'
    first_name: 'Jane'
    middle_initial: 'A'
    email: 'jane_doe@gmail.com'
    organization: 'NIH'
    department: 'NCBI'
    phone: '301-555-0245'
    fax: '301-555-1234'
    street: '9000 Rockville Pike'
    city: 'Bethesda'
    state: 'MD'
    postal_code: '20850'
    country: 'USA'
authors:
    - author:
        last_name: 'Doe'
        first_name: 'Jane'
        middle_initial: 'A'
    - author:
        last_name: 'Doe'
        first_name: 'John'
bioproject: 'PRJ9999999'
biosample: 'SAMN99999999'
locus_tag_prefix: 'tmp'
publications:
    - publication:
        pmid: 29112715
```

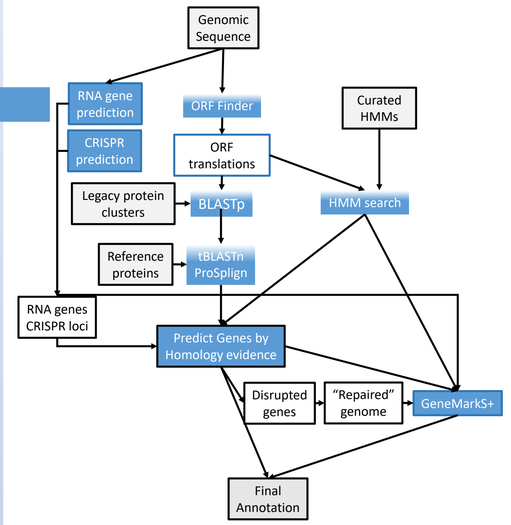
Stand-alone PGAP

- ✓ Conforming with PGAP at NCBI
- ✓ Without dependencies on NCBI resources
- ✓ Reproducible
- ✓ Producing data acceptable to GenBank
- ⚒ Executable by external users on a variety of platforms



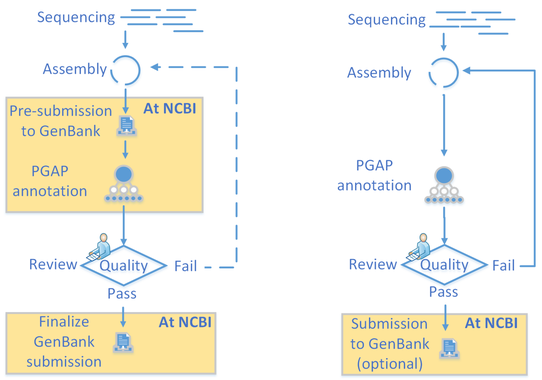
The PGAP annotation pipeline at NCBI

- Used for the annotation of over 170,000 GenBank and 158,000 RefSeq genomes
- Written in an in-house workflow language
- Optimized for NCBI's internal computational resources



Flow of assemblies from individual labs to GenBank

Before: submit and request annotation ... **Now**: annotate, then submit



Future work

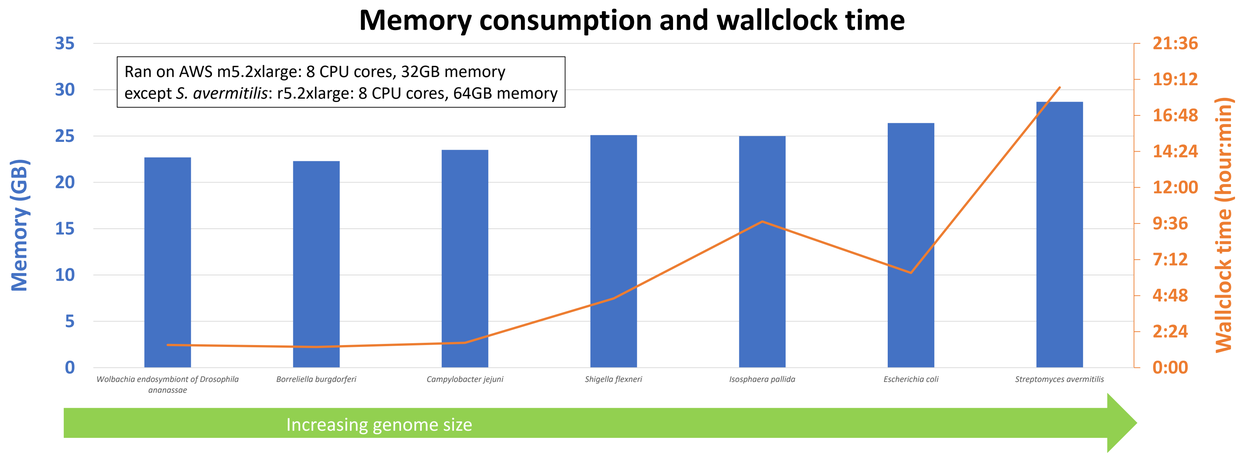
- ⚒ Add taxonomy check (Average Nucleotide Identity)
- ⚒ Test on additional platforms (Arvados, Toil, etc.)
- ⚒ Improve performance

Getting started

1. Download the script
   `curl -OL https://github.com/ncbi/pgap/raw/prod/scripts/pgap.py`
2. Download the Docker image, the reference data, the cwl code, and cwltool
   `./pgap.py --update`
3. Annotate your favorite genome
   `./pgap.py -r -o my_genome.results my_genome.yaml`

Or download the Docker image, reference data and CWL code separately and integrate into your own CWL platform (see: https://github.com/ncbi/pgap/wiki/Installation)

Memory consumption and wallclock time



Frequently Asked Questions

- **Does Docker need to be installed on my machine?**

Yes.

- **Do I need to be able to execute in Docker?**

Yes. Please ensure you can successfully execute the following command to test your configuration:
`$ docker run hello-world`

- **How much space and memory do I need?**

You need about 100GB of storage space from the supplemental data and working space, and 30GB of RAM

- **Can I run under MacOS or Windows?**

Although Linux is our primary development platform, we do support running under other operating systems using `pgap.py`. You will still need Python 3.5 or greater, and Docker. Ensure that Docker is running with Linux support enabled.

- **Do I need network access?**

Yes - currently several programs require network access. We are working on lifting this requirement

- **Why does my run occasionally not finish, producing no logs or message in terminal, and yet the pipeline still seem to be running?**

You are most likely running the pipeline on a remote machine over `ssh`, and the connection has been interrupted. Use the `nohup` utility, or a terminal multiplexer, such as `tmux` or `screen` when working on a remote machine, to allow `pgap.py` to continue in case the `ssh` connection is interrupted.

U.S. National Library of Medicine
National Center for Biotechnology Information

This work was supported by the Intramural Research Program of the National Library of Medicine, National Institutes of Health. *(F. Thibaud-Nissen, D. Slotta, A. Badretdin, B. Busby, R. Cohen, W. Li, W. Hlavina)*